Requirements Engineering

Hardware\_SYS

Submitted By: Jack O’Hara (T00195057)

Computing with Multi Media

Date Submitted: dd/mm/yyyy

**Table of Contents**

[1. Introduction/overview 3](#_Toc474162299)

[2. Functional Components 4](#_Toc474162300)

[3. User Requirements 5](#_Toc474162301)

[4. System Requirements 6](#_Toc474162302)

[4.1. System Level Use Case Diagram 6](#_Toc474162303)

[4.2. Module 1 (Use active Verb + Noun) 6](#_Toc474162304)

[4.2.1. Functional Requirement 1 (Use active verb + noun) 6](#_Toc474162305)

[4.2.2. Functional Requirement 2 (Use active verb + noun) 6](#_Toc474162306)

[4.2.3. Functional Requirement 3 (Use active verb + noun) 6](#_Toc474162307)

[4.3. Module 2 (Use active Verb + Noun) 6](#_Toc474162308)

[4.3.1. Functional Requirement 1 (Use active verb + noun) 6](#_Toc474162309)

[4.3.2. Functional Requirement 2 (Use active verb + noun) 6](#_Toc474162310)

[4.4. Module 3 (Use active Verb + Noun) 6](#_Toc474162311)

[4.4.1. Functional Requirement 1 (Use active verb + noun) 6](#_Toc474162312)

[4.4.2. Functional Requirement 2 (Use active verb + noun) 6](#_Toc474162313)

[5. System Model 7](#_Toc474162314)

[5.1. Level-0 DFD 7](#_Toc474162315)

[5.2. Level-1 DFD 7](#_Toc474162316)

[5.3. Level-2 DFD (Process P1: Title) 7](#_Toc474162317)

[5.4. Level-2 DFD (Process P2: Title) 7](#_Toc474162318)

[5.5. Level-2 DFD (Process P3: Title) 7](#_Toc474162319)

[6. Data Model (Class Diagram) 8](#_Toc474162320)

[6.1. Class Diagram 8](#_Toc474162321)

[6.2. Relational Schema 8](#_Toc474162322)

[6.3. Database Schema 8](#_Toc474162323)

[7. Conclusion 9](#_Toc474162324)

[8. Appendices 10](#_Toc474162325)

[8.1. Appendix A – Title 10](#_Toc474162326)

[8.2. Appendix B – Title 10](#_Toc474162327)

# Introduction/overview

In this assignment, I hope to explain the functionality of my system. I have chosen to create a system for a hardware store. The reason I have chosen this is because I am a part time employee at Kelliher’s Hardware store Tralee. In the store there are no computing systems involved. None of the stock have bar codes, so everything must be added up by calculator. The only way to do a stock check is by counting all the goods in the shop and as you can imagine, this can be very tedious. We have an invoice book for customers which allows them to purchase goods on credit. The customer can come into the shop whenever to pay off they’re docket. In the store we have one room filled with invoice books which is taking up valuable space in the shop. When making sales no information is given on the receipt except the final sale price, time and date, this can cause some problems as it is hard to know if the customer is being fully truthful with you. When I was told that my requirement engineering project was to be based on a transactional system Kelliher’s immediately came to my mind. My system will allow the user to check the stock which is available in the store with just a few clicks, store all customer details and all they’re purchases without taking up any room in the shop. It will also check exactly what the customer has purchased using they’re Sale ID. I’m sure there is many other Hardware stores out there that have not yet progressed into the computing age so I believe my system will have a big audience.

# Functional Components

# User Requirements

1 HardwareSys will deal with Stock

* 1. HardwareSys will register new stock
  2. HardwareSys will update stock
  3. HardwareSys will Discontinue stock
  4. HardwareSys will perform a stock enquiry

2HardwareSys will deal withCustomers

2.1HardwareSys will register a Customer

2.2HardwareSys will update Customer details

2.3HardwareSys will Query Customer details

3HardwareSys will perform Sales

3.1HardwareSys will register a sale

3.2HardwareSys will return a sale

3.3HardwareSys will issue an invoice

3.4HardwareSyswill pay an invoice

4HardwareSys will act as a SalesAdmin

4.1HardwareSys will perform sales analysis

4.2HardwareSys will perform stock analysis

# System Requirements

## System Level Use Case Diagram

HardwareSYS

Staff

Customer

## Stock

My System deals with Stock. It allows you to register stock, Update stock, discontinue stock and enquire an item of stock

### Register New Stock

This function records the details of a new stock item available for sale in the system so that the item can go on sale.

|  |  |
| --- | --- |
|  |  |
| Staff | System |
| Invoke Register Stock function  Enters Details  valid  no  yes  Assign status Avalable  Saves details  Displays Confirmation Message  Clear UI | Validates data  Display UI  Retrieve Stock Department  Generate next Stock\_ID |

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register New Stock** | |
| **Use Case Id** | 4.1.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function records the details of a new stock item available for sale in the system. | |
| **Preconditions** |  | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Staff** | **System Response** |
|  | **Step 1:**invokes the new stock function  **Step 5:**User enters the details for the new stock:   * Department ID * Name, * Qty, * Cost\_Price, * Sale\_Price | **Step 2:**System generates the next Stock ID  **Step 3:** System retrieves stock Department from the Department File and loads in form control  **Step 4:** System displays the UI  **Step 6:**System Validates the details:   * All fields must be entered * Stock\_Name must not contain numeric values only * Qty, Cost\_Price and Sale\_Price must only contain numeric values * Sale\_Price has to be greater than Cost\_Price   **Step7:** System assigns a default status of ‘Available’  **Step 8:** The System saves details to the stock file:   * Stock ID * Department ID * Name, * Qty, * Cost\_Price, * Sale\_Price * Status   **Step9:** Display a conformation message.  **Step10:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Not all fields entered** |  | **Step6:** A blank field has been detected by the system  **Step7:** System displays an error message “not all fields have been entered ”  **Step8:** Retuns the user to Step 5 |
| **Stock\_Name was entered with numeric values only** |  | **Step6:** System detected that the user entered Stock\_Name with numeric values only  **Step7:** System displays an error message “Stock\_Name must not contain numeric values only”  **Step8:** Returns the user to Step5 |
| **Qty, Cost\_Price and Sale\_Price was not entered with numeric values only** |  | **Step6:** System detected that the user entered Qty,Cost\_Price or Sale\_Price was not entered with numeric values only  **Step7:** System displays an error message “Qty, Cost\_Price and Sale\_Price must only contain numeric values”  **Step8:** Returns the user to Step5 |
| **Cost\_Price is greater than Sale\_Price** |  | **Step6:** System detected that the user entered Cost\_Price greater than Sale\_Price  **Step7:** System displays an error message “Sale\_Price has to be greater than Cost\_Price”  **Step8:** Returns the user to Step5 |
| **Conclusions** | Stock will be added to the stock file | |
| **Post conditions** | When user requires a Stock Enquiry the stock that the user requires information onwill appear in the file | |
| **Business Rules** | Cannot register stock if it has already been registered, Sale\_Price has to be greater than Cost\_Price | |
| **Implementation Constraints** |  | |

### Update Stock

This Function allows the user to update an item of Stock in the System

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Stock** | |
| **Use Case Id** | 4.1.2 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** | Staff | |
| **Description** | This function allows the user to update an item of stock from Hardware\_SYS | |
| **Preconditions** | The stock that the user is trying to update has to be in the system already | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Update\_Stock function  **Step 3:**User enters Stock Name  **Step 6 :**User selects the Stock they wish to edit    **Step 8:** The User enters the new details about the stock | **Step 2:**System displays the UI    **Step 4:**The System validates the data :   * Field is not left blank * Stock Name must related to Stock in the system   **Step 5:**the System displays all stock with a similar name to the stock name entered.  **Step 7:**System retrieves all data in relation to the stock selected and displays them in editable text boxes  **Step 9:**The System validates the details   * Stock\_Name must not contain numeric values only * Qty, Cost\_Price and Sale\_Price must only contain numeric values * Sale\_Price has to be greater than Cost\_Price   **Step10**The System updates the new details in the Stock file    **Step11:** Displays the conformation message.  **Step12:** Clears UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step4:** A blank field has been detected by the system  **Step5:** System displays an error message “Stock ID has been left blank ”  **Step6:** Returns the user to Step3 |
| **Stock Name does not relate to any stock in the system in the System** |  | **Step4:** System detected that the user entered Stock name does not relate to any stock in the system in the system  **Step5:** System displays an error message “That Stock Name does not exist”  **Step6:** Returns the user to Step 3 |
| **Not all fields entered** |  | **Step9:** A blank field has been detected by the system  **Step10:** System displays an error message “not all fields have been entered ”  **Step11:** Returns the user to Step6 |
| **Stock\_Name was entered with numeric values only** |  | **Step9:** System detected that the user entered Stock\_Name with numeric values only  **Step10:** System displays an error message “Stock\_Name must not contain numeric values only”  **Step11:** Returns the user to Step6 |
| **Qty, Cost\_Price and Sale\_Price was not entered with numeric values only** |  | **Step9:** System detected that the user entered Qty, Cost\_Price or Sale\_Prices was not entered with numeric values only  **Step10:** System displays an error message “Qty, Cost\_Price and Sale\_Price must only contain numeric values”  **Step11:** Returns the user to Step6 |
| **Cost\_Price was greater than Sale\_Price** |  | **Step9:** System detected that the user entered Cost\_Price greater than Sale\_Price  **Step10:** System displays an error message “Sale\_Price has to be greater than Cost\_Price”  **Step11:** Returns the user to Step6 |
| **Conclusions** | Stock in the Stock table will be updated | |
| **Post conditions** | When user requires a Stock Enquiry the stock updated will appear in the file | |
| **Business Rules** | Stock cannot be updated if it has been discontinued | |
| **Implementation Constraints** |  | |

### 4.2.3. Discontinue Stock

This Function allows the user to Discontinue an item of Stock From the System

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Discontinue Stock** | |
| **Use Case Id** | 4.1.3 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** | Staff | |
| **Description** | This function allows the user to discontinue an item of stock from Hardware\_SYS | |
| **Preconditions** | The stock that the user is trying to discontinue has to be in the system already | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Discontinue Stock function  **Step 3:**User enters Stock Name  **Step 6:**User selects the Stock they wish to discontinue | **Step 2:**System displays the UI  **Step 4:**The System validates the data :   * Field is not left blank * Stock Name must related to Stock in the system   **Step 5:**the System displays all stock with a similar name to the stock name entered.  **Step 7:** The System will change the stocks status from A to U  **Step8:** Displays the conformation message.  **Step9:** Clears UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step4:** A blank field has been detected by the system  **Step5:** System displays an error message “Stock ID has been left blank ”  **Step6:** Returns the user to Step3 |
| **Stock Name does not relate to any stock in the system in the System** |  | **Step4:** System detected that the user entered Stock name does not relate to any stock in the system in the system  **Step5:** System displays an error message “That Stock Name does not exist”  **Step6:** Returns the user to Step 3 |
| **Conclusions** | Stocks status will have been updated to Unavailable | |
| **Post conditions** | When user requires a Stock Enquiry the stock updated will appear in the file | |
| **Business Rules** | Stock cannot be discontinued unless it is already in the file. | |
| **Implementation Constraints** |  | |

### Stock Enquiry

This Function allows the user to view all the Stock registered in the shop

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Stock Enquiry** | |
| **Use Case Id** | 4.1.4 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the user to see all information on the stock which they are selling | |
| **Preconditions** |  | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Stock\_Enquiry function  **Step 3:**User enters Stock Name  **Step 6:**User selects the Stock they wish to view details on | **Step 2:**System displays the UI  **Step 4:**The System validates the data :   * Field is not left blank * Stock Name must relate to Stock in the system   **Step 5:** the System displays all stock with a similar name to the stock name entered.  **Step 7:** The System retrieves all data from the Stock file in relation to the stock id  **Step 8:** The System Displays   * Stock Name * Department * Qty * Cost Price * Sale Price * Status   of the Selected Stock  **Step9:** Clears UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step4:** A blank field has been detected by the system  **Step5:** System displays an error message “Stock ID has been left blank ”  **Step6:** Returns the user to Step3 |
| **Stock Name does not relate to any stock in the system in the System** |  | **Step4:** System detected that the user entered Stock name does not relate to any stock in the system in the system  **Step5:** System displays an error message “That Stock Name does not exist”  **Step6:** Returns the user to Step 3 |
| **Conclusions** | Stock will be displayed to the manager | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Customers

My System deals with Customers. It allows the user to register a Customer, update customer details and query customer details

### Register Customer

This function allows the user to register a Customer in the system in order for them to purchase stock on credit

Staff

Customer

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register Enquiry** | |
| **Use Case Id** | 4.2.1 | |
| **Priority** | Medium | |
| **Source** | Customer | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the user to Register a Customer to the System | |
| **Preconditions** |  | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Register Customer function  **Step 4:**User enters the details for the Customer   * Forename * Surname * Number * Address Line 1 * Address Line 2 * Address Line 3 | **Step 2:**System generates the next Customer\_ID  **Step 3:** System displays the UI  **Step 5:**System Validates the details:   * All fields must be entered * Forename and Surname must not contain numeric values * Number must only contain numeric values * All of the address lines must contain at least one alphabetical character * User enters details that is already in the file   **Step 6:** The System saves details to the customer file:   * Customer\_ID * Forename * Surname * Number * Address Line 1 * Address Line 2 * Address Line 3   **Step9:** Display a conformation message.  **Step10:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step5:** A blank field has been detected by the system  **Step6:** System displays an error message “All Fields must be entered”  **Step7:** Returns the user to Step4 |
| **Forename and Surname contained numeric values** |  | **Step5:** The System detected that Forename or Surname contains numeric values  **Step6:** System displays an error message “Forename and Surname cannot contain numeric values”  **Step7:** Returns the user to Step4 |
| **Number did not only contain numeric values** |  | **Step5:** The System detected that Number contains non-numeric values  **Step6:** System displays an error message “Number can only contain numeric values”  **Step7:** Returns the user to Step4 |
| **At least one of the address lines did not contain at least one alphabetical character** |  | **Step5:** The System detected that one or more of the addressdoes not containalphabeticalcharacters  **Step6:** System displays an error message “Address lines must contain at least one or more alphabetical characters”  **Step7:** Returns the user to Step4 |
| **User enters details that is already in the file** |  | **Step5:** The System detected that details entered in the system are identical to a customer that is already registered  **Step6:** System displays an error message “This customer is already registered ”  **Step7:** Returns the user to Step4 |
| **Conclusions** | Customer will be registered | |
| **Post conditions** | Customer will be allowed to purchase items on credit | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Update Customer

This function allows the user to update information about customers that are registered in the system

Staff

Customer

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Customer** | |
| **Use Case Id** | 4.2.2 | |
| **Priority** | Medium | |
| **Source** | Customer | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the user to Update a Customers details on the System | |
| **Preconditions** | The Customer must already be registered on the system | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Update Customer function  **Step 3:**User enters Customer Surname  **Step 5 :**User selects the Customer they wish to edit    **Step 7:** The User enters the new details about the stock | **Step 2:**System displays the UI  **Step 4:**the System displays all Customers with a similar surname to the surname entered.  **Step 6:**System retrieves all data in relation to the Customer selected and displays them in editable text boxes  **Step 8:**The System validates the details   * All fields must be entered * Forename and Surname must not contain numeric values * Number must only contain numeric values * All of the address lines must contain at least one alphabetical character * User enters details that is already in the file   **Step9**The System registers the new details in the Stock file    **Step10:** Displays the conformation message.  **Step11:** Clears UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step8:** A blank field has been detected by the system  **Step9:** System displays an error message “Customer\_ID was left blank”  **Step10:** Returns the user to Step7 |
| **Forename and Surname contained numeric values** |  | **Step8:** The System detected that Forename or Surname contains numeric values  **Step9:** System displays an error message “Forename and Surname cannot contain numeric values”  **Step10:** Returns the user to Step7 |
| **Number must only contain numeric values** |  | **Step8:** The System detected that Number contains non-numeric values  **Step9:** System displays an error message “Number can only contain numeric values”  **Step10:** Returns the user to Step7 |
| **All of the address lines must contain at least one alphabetical character** |  | **Step8:** The System detected that one or more of the address does not contain alphabetical characters  **Step9:** System displays an error message “Address lines must contain at least one or more alphabetical characters”  **Step10:** Returns the user to Step7 |
| **User enters details that is already in the file** |  | **Step8:** The System detected that details entered in the system are identical to a customer that is already registered  **Step9:** System displays an error message “This customer is already registered ”  **Step10:** Returns the user to Step7 |
| **Conclusions** | Customer details will be updated | |
| **Post conditions** | Customer details will appear different in the system | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Query Customer

This function allows the user to see all information about any customer registered in the system

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Query Customer** | |
| **Use Case Id** | 4.2.3 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the user to see all information on a customer | |
| **Preconditions** |  | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Query Customer function  **Step 3:**User enters Customer Surname  **Step 6:**User selects the Customer they wish to view details on | **Step 2:**System displays the UI  **Step 4:**The System validates the data :   * Field is not left blank * Surname entered must relate to a customer in the system   **Step 5:**the System displays all customers with a similar surname to the surname entered.  **Step 7:** The System retrieves all data from the Customer file in relation to the Customer selected  **Step 8:** The System Displays   * Customer\_ID * Forename * Surname * Number * Address Line 1 * Address Line 2 * Address Line 3   Of the selected customer  **Step9:** Clears UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step4:** A blank field has been detected by the system  **Step5:** System displays an error message “Stock ID has been left blank ”  **Step6:** Returns the user to Step3 |
| **Surname does not relate to any Customer in the system in the System** |  | **Step4:** System detected that the user entered Surname does not relate to any Customer in the system  **Step5:** System displays an error message “That Surname does not exist in this system”  **Step6:** Returns the user to Step 3 |
| **Conclusions** | Customer information will be displayed to the manager | |
| **Post conditions** | Customer must be registered in the system before an enquiry can be made on him/her | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Sales

My project will deal with Sales. It will allow the user to register sales, return items, issue invoices, pay invoices

### Register Sale

This function records the details of a new stock item available for sale in the system.

Customer

Staff

|  |  |
| --- | --- |
| Activity Diagram: Register Sale | |
| Staff | System |
| Enter Sale Details  Invoke Register Sale Function | Generate the next Sale\_ID  System retrieves Department form the Department file  Display UI  Validate Sale Details  Error Message  No  Yes  valid  Retrieves sale price from the stock file  System calculates final\_sale\_price  Displays Final\_Sale\_Price  System retrieves payment type form the sale file |

|  |  |
| --- | --- |
|  | Reset UI  Qty gets updated in the stock file  Saves Details in the Sale file |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register Sale** | |
| **Use Case Id** | 4.3.1 | |
| **Priority** | High | |
| **Source** | Customer | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function records the details of a new stock item available for sale in the system. | |
| **Preconditions** |  | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Staff** | **System Response** |
|  | **Step 1:**invokes the Register Sale function  **Step 4:**User enters Stock Name  **Step 7 :**User selects the Stock they wish to add to the cart and then modifies the cart and clicks the register sale button when finished  **Step 9 :**User selects cash sale or credit sale  **Step 11:**User enters Customer Surname  **Step 13 :**User selects the Customer they wish to edit  **Step 15 :**User presses the register sale button | **Step 2:**System generates the next Sale\_ID  **Step 3:** System displays UI  **Step 5:**The System validates the data :   * Field is not left blank * Stock Name must related to Stock in the system   **Step 6:** the System displays all stock with a similar name to the stock name entered.  **Step 8:** The System displays radio buttons to see if the user wants to do a cash sale or a credit sale  **Step 10:** if credit sale is selected the system displays a customer search box an    **Step 12:**System retrieves all data in relation to the Customer selected and displays them in editable text boxes  **Step 14 :**System displays the register sale button  **Step 16:** The System saves details to the sale file:   * Sale\_ID * Status * Sale date * And Customer\_ID if credit is selected   **Step 17:** The System saves details to the saleitems file:   * Sale\_ID * Stock id * Qty sold * Sale price   **Step 18:** Qty in the Stock file for the Stock\_ID entered gets subtracted by Qty\_Sold entered  **Step 19:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Not all fields entered** |  | **Step5:** A blank field has been detected by the system  **Step6:** System displays an error message “not all fields have been entered ”  **Step7:** Returns the user to Step 4 |
| **Stock\_Name was entered with numeric values only** |  | **Step5:** System detected that the user entered Stock\_Name with numeric values only  **Step6:** System displays an error message “Stock\_Name must not contain numeric values only”  **Step7:** Returns the user to Step4 |
| **Stock name does not exist in the sale file** |  | **Step5:** no stock was returned from the search  **Step6:** System displays an error message “that stock does not exist in the system ”  **Step7:** Retuns the user to Step 4 |
| **Conclusions** | Sale will be registered | |
| **Post conditions** | If Customer wishes to return an item of stock the Sale\_ID will be required | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Return Item

This function records the details of an item that a customer is returning.

Staff

Customer

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Return Item** | |
| **Use Case Id** | 4.3.2 | |
| **Priority** | High | |
| **Source** | Customer | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** | Source or destination of information…… | |
| **Description** | This function records the details of an item that the customer is returning. | |
| **Preconditions** | The item has to have been bought from the shop with a Sale\_ID | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Staff** | **System Response** |
|  | **Step 1:**invokes the Return Item function  **Step 4:**User enters data sale id  **Step 7:** The user selects which item of stock they wish to return if it will go back into stock and the qty they will be returning | **Step 2:** System retrieves Refund Type from the Refund Types File and loads in form control  **Step 3:** System displays the UI  **Step 5:**System Validates the details:   * Sale\_ID must have taken place on the system * Sale\_ID must not be blank * Sale\_id has to be numeric * Sale\_id cannot have all its items returned   **Step 6:** The system retrieves all items with similar sale id and displays it in a data grid box  **Step8** The system validates if the qty that is entered is greater than the qty they wish to return  **Step9:** If Item back in Stock is selected yes Qty of the Stock\_ID get plus Qty\_Sold in the Stock file.  **Step 10:** sets the sale price to zero in the saleitems file  **Step 11:** Conformation message appears  **Step12:** Clear UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Sale\_ID was left blank** |  | **Step5:** A blank field in Sale\_ID has been detected by the system  **Step6:** System displays an error message “Sale\_ID has been left blank ”  **Step7:** Returns the user to Step4 |
| **Sale\_ID without numeric values** |  | **Step5:** System detected that the user entered Sale\_ID without numeric values only  **Step6:** System displays an error message “Sale\_ID must contain numeric values only”  **Step7:** Returns the user to Step4 |
| **Sale id did not take place in the system** |  | **Step5:** System detected that the Sale id did not take place on this system  **Step6:** System displays an error message “that sale did not take place on the system”  **Step7:** Returns the user to Step4 |
| **All items from the sale id has been returned** |  | **Step5:** System detected that all the items from the sale id has been returned  **Step6:** System displays an error message “all items from the sale id has been returned”  **Step7:** Returns the user to Step4 |
| **Qty sold is less than qty returned** |  | **Step8:** System detected that the qty returned is greater than the qty sold  **Step9:** System displays an error message “quantity returned has to be less than qty sold”  **Step10:** Returns the user to Step7 |
| **Conclusions** | Item will be refunded | |

### Issue Invoice

This function displays a Customers invoice.

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Issue Invoice** | |
| **Use Case Id** | 4.3.3 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the user to see all information on a Customers Purchases. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Issue\_Invoice function  **Step 3:**User enters Customer Surname  **Step 6:**User selects the Customer they wish to Issue an Invoice to | **Step 2:**System displays the UI  **Step 4:**The System validates the data :   * Field is not left blank * Surname entered must relate to a customer in the system   **Step 5:**the System displays all customers with a similar surname to the surname entered.  **Step 7:**The system retrieves data from the sale file in relation to the customer id  **Step 8:**The system retrieves data from the sale items file in relation to the sale id  **Step 9:**The system retrieves address of the customer from the customer file in relation to the customer id  **Step 10:** The System multiplies the Sale\_Price by the Qty\_Sold to every line in the file in which cutomer\_id is present  **Step 11:**The System adds it all up to get Final sale  **Step 12:** The System Displays a table of   * Sale\_ID * Stock\_ID * Stock Name * Sale Price * Qty * Total sale price * Sale date   Of each item of Sale that the Customer has purchased  Underneath the table will show the Final\_Cost and the customers address to send the invoice to  **Step13:** Clears UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step4:** A blank field has been detected by the system  **Step5:** System displays an error message “Stock ID has been left blank ”  **Step6:** Returns the user to Step3 |
| **Surname does not relate to any Customer in the system in the System** |  | **Step4:** System detected that the user entered Surname does not relate to any Customer in the system  **Step5:** System displays an error message “That Surname does not exist in this system”  **Step6:** Returns the user to Step 3 |
| **Conclusions** | Customer purchases will be displayed to the Staff | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Pay Invoice

This function allows a Customers to pay his/her invoice.

Customer

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Pay Invoice** | |
| **Use Case Id** | 4.3.4 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | This function allows a customer to pay his/her Invoice | |
| **Preconditions** |  | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Pay\_Invoice function  **Step 3:**User enters Customer Surname  **Step 6:**User selects the Customer they wish to Issue an Invoice to  **Step 13:** user presses pay now button | **Step 2:**System displays the UI  **Step 4:**The System validates the data :   * Field is not left blank * Surname entered must relate to a customer in the system   **Step 5:**the System displays all customers with a similar surname to the surname entered.  **Step 7:**The system retrieves data from the sale file in relation to the customer id  **Step 8:**The system retrieves data from the sale items file in relation to the sale id  **Step 9:**The system retrieves address of the customer from the customer file in relation to the customer id  **Step 10:** The System multiplies the Sale\_Price by the Qty\_Sold to every line in the file in which cutomer\_id is present  **Step 11:**The System adds it all up to get Final sale  **Step 12:** The System Displays a table of   * Sale\_ID * Stock\_ID * Stock Name * Sale Price * Qty * Total sale price * Sale date   Of each item of Sale that the Customer has purchased  Underneath the table will show the Final\_Cost  **Step 14:** the staus of each sale gets set to (p) payed  **Step15:** Clears UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step4:** A blank field has been detected by the system  **Step5:** System displays an error message “Stock ID has been left blank ”  **Step6:** Returns the user to Step3 |
| **Surname does not relate to any Customer in the system in the System** |  | **Step4:** System detected that the user entered Surname does not relate to any Customer in the system  **Step5:** System displays an error message “That Surname does not exist in this system”  **Step6:** Returns the user to Step 3 |
| **Conclusions** | A Customer will Pay off his/her invoice | |

## Sales Admin

My System will allow the user to make a Sale Analysis and make a Stock Analysis.

### Sales Analysis

This function displays a Sale Analysis.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Sales Analysis** | |
| **Use Case Id** | 4.4.1 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the user to see all information on Sales throughout the shop | |
| **Preconditions** | Sales have to be made in order to use this function | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Sales Analysis function  **Step5 :**User enters the year and if they would like to search by customer or all sales  **Step 5:**user enters a surname | **Step 2:**System displays the UI  **Step 4:** if customer is selected the user is asked to enter a customer surname to search  **Step6:**System validates that the surname is :   * Alphabetic only * Exists on the system * Not blank   **Step6:** System retrieves Sale\_ID, Sale Value, Status and Sale date from the Sales file in relation to the year and or customer id entered it then displays a chart with this data |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field was left blank** |  | **Step6:** A blank field has been detected by the system  **Step7:** System displays an error message “Customer Surname has been left blank ”  **Step8:** Returns the user to Step5 |
| **Surname does not exist on the system** |  | **Step6:** no rows were added to the table  **Step7:** System displays an error message “that surname does not exist on the system ”  **Step8:** Returns the user to Step5 |
| **Surname was not entered with alphabetic characters only** |  | **Step6:** A blank field has been detected by the system  **Step7:** System displays an error message “Customer Surname has been entered without alphabetic characters only ”  **Step8:** Returns the user to Step5 |
| **Conclusions** | Sales will be displayed to the manager | |
| **Post conditions** | Sales will have to be registered in order for this function to work | |
| **Implementation Constraints** |  | |

|  |  |  |
| --- | --- | --- |
| Sale\_ID | Customer\_ID | Final Sale Price |
|  |  |  |

### Stock Analysis

This function displays a Stock Analysis.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Stock Analysis** | |
| **Use Case Id** | 4.4.2 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function allows the user to see all information on Stock throughout the shop | |
| **Preconditions** | Sales have to be made in order to use this function | |
| **Trigger** | Is this Use Case invoked by another Use Case??? | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**invokes the Stock Analysis function  **Step5 :**User enters the Department and year | **Step 2:**System displays the UI  **Step 3:** System retrieves Department from the Department file and loads in form control  **Step 4:**The system asks the user to enter Department in which they would like to see and which year.  **Step 6:**System retrieves data from the Stock file and Sale file in relation to the Department  **Step 9:** The system displays a table of   * Stock\_ID * Stock Name * Cost\_Price * Sale\_Price * Number\_Of\_Sales   Of each stock item in the selected department and a chart indicating the number of sales  **Step10:** Clears UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Conclusions** | Sales will be displayed to the manager | |
| **Post conditions** | Sales have to be made in order for this function to work | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stock\_ID | Stock Name | Cost\_Price | Sale\_Price | Qty Sold |
|  |  |  |  |  |

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

Details

purchase

Hardware\_SYS

staff

customer

Details

cost

## Level-1 DFD

Sale Details

Customer

Stock details

P1

Process

Stock

P3

Process

Sales

Stock Details

Sale details

D1

Stock File

Customer details

P4

Process Analysis

P2

Process Customers

D2

Sale File

Stock details

Customer details

D3

Customer File

Sale details

D1

Stock File

Customer Details

Customer

D2

Sale File

## Level-2 DFD (Process P1: Stock)

Stock Status

P1.1

Register

Stock

P1.3

Discontinue

Stock

Stock Details

D1

Stock File

Stock Details

Stock Details

Stock Details

P1.4

Stock

Enquiry

P1.2

Update

Stock

## Level-2 DFD (Process P2: Customer)

Customer Details

Customer

Customer Details

P2.1

Customer Details

P2.2

Update Customer

Register Customer

Customer Details

D3

Customer File

Customer Details

P2.3

Query Customer

## Level-2 DFD (Process P3: Sale)

Customer

Sale price

Refund price

Sale Details

Sale Details

Sale Details

P3.2

Return Item

P3.1

Register Sale

Sale Details

**D2**

**Sale File**

Stock Details

Sale Details

Stock Details

Stock Details

D1

Stock File

Customer Details

Customer Details

D3

Customer File

Customer Details

Payment Details

Customer Details

**D2**

**Sale File**

Sale Details

Payment Details

P3.4

Pay Invoice

P3.3

Issue Invoice

Customer

Sale Details

## Level-2 DFD (Process P4: Sale Admin)

P4.1P2.1

Sales Analysis Register Customer

Stock Details

Sale Details

Sale Details

**D2**

**Sale File**

D1

Stock File

P4.2

Stock Analysis

# Data Model (Class Diagram)

My Data Model gives a brief overview of all the files that will be in my system and how the data will be saved in relation to the files.

## Class Diagram



## Relational Schema

Stock (Stock\_ID, Stock Name, Qty, Cost Price, Sale Price, Status, DepartmentID)

Department (Department ID, DeptName)

Customer (Customer\_ID, Forename, Surname, Address line 1, Address line 2, Address line 3)

Sale (Sale ID, SaleDate, Status, CustomerID)

Saleitems (SaleID, StockID, Qty Sold, Unit Price)

## Database Schema

**Schema:** **HardWare Sys**

**Relation: Stocks**

Attributes:

Stock\_ID numeric(4) NOT NULL,

Stock\_Name varchar2 (30) NOT NULL,

Qty numeric (4) NOT NULL,

Cost\_P decimal(6,2) NOT NULL,

Sale\_P decimal(6,2) NOT NULL,

Status varchar2 (1) NOT NULL,

Department\_ID numeric(1) NOT NULL,

**Primary Key:** Stock ID

[**Foreign Key:** Department ID reference Department]

**Relation: Departments**

Attributes:

Department\_ID numeric(1) NOT NULL,

Department\_Name varchar2 (10) NOT NULL,

**Primary Key**: (Department\_ID)

**Relation: Customers**

Attributes:

Customer ID numeric(4) NOT NULL,

Forename varchar2(10) NOT NULL,

Surname varchar2 (10) NOT NULL,

Address Line 1 varchar2(10) NOT NULL,

Address Line 2 varchar2 (10) NOT NULL,

Address Line 3 varchar2 (10) NULL,

**Primary Key:** Customer ID

**Relation: Sales**

Attributes:

Sale\_ID numeric(6) NOT NULL ,

Sale\_Date Date NOT NULL,

Status varchar2 (1) NOT NULL,

Customer\_ID numeric(4) NULL,

Sale\_Value decimal(6,2) NULL,

**Primary Key:** Sale\_ID

[**Foreign Key:** Customer ID reference Customer]

**Relation: Sale Items**

Attributes:

QtySold numeric(3) NOT NULL ,

Price decimal (4,2) NOT NULL,

Sale\_ID numeric(6) NOT NULL,

Stock\_ID numeric (3) NOT NULL,

[**Foreign Key:** Stock ID reference Stock]

[**Foreign Key:** Sale ID reference Sale]

# Conclusion

# Appendices

## Appendix A – Title

## Appendix B – Title

Might include:

* **Lookup / Referencetables**
* **Sample reports / Listings**